

# CS368 MATLAB Programming

## Lecture 10

Young Wu

Based on lecture slides by Michael O'Neill and Beck Hasti

April 6, 2022

# Indicator Variables

## Math

- If the same task is performed for different values of a variable, use an indicator variable and vectorize.
- If different tasks are performed for different values of a variable, use a *switch* conditional.
- If different tasks are performed under different conditions, use an *if* conditional.

# Switch

## Code

- Different tasks are performed for  $x = v_1$ , for  $x = v_2$  or  $v_3$ , and for every other value of  $x$ .

```
1 switch x
2   case v1
3     ...
4   case {v2, v3}
5     ...
6   otherwise
7     ...
8 end
```

# If Else

## Code

- Different tasks are performed if  $x \neq 0$ , if  $x = 0$  but  $y \neq 0$ , and if  $x = 0$  and  $y = 0$ .

- 1 *if*  $x$
- 2 ...
- 3 *elseif*  $y$
- 4 ...
- 5 *else*
- 6 ...
- 7 *end*

# Condition for If

## Code

- *if*  $x$  and *if*  $x \sim= 0$  represent the same condition. The expression  $x \sim= 0$  should be treated as a variable whose value is  $\begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$ .
- *while*  $x$  and *while*  $x \sim= 0$  represent the same loop for the same reason.

# Conditionals Quiz Questions

## Quiz

# Number of Input Arguments

## Code

- When the function *function*  $z = f(x, y)$  is called, 0, 1 or 2 arguments can be provided.
- *switch* can be used here to perform different tasks when different number of arguments are given.
- *nargin* is the number of input arguments provided when the function is called.

# Log with Optional Input Arguments

## Code

- For example, a new log function can be defined by  $\text{log}()$  returns 1,  $\text{log}(x)$  returns natural  $\log(x)$ , and  $\text{log}(n, x)$  returns  $\log_n(x)$ .

```
1 function z = log(x, y)
2     switch nargin
3         case 1
4             z = log(x);
5         case 2
6             z = log(y) / log(x);
7         otherwise
8             z = 1;
9     end; end
```



# Variable Length Input Argument

## Code

- *varargin* represents an arbitrary number of input variables.
- It can only be used as the last argument of a function, for example, *function y = f(x1, x2, x3, varargin)*.
- The *i*-th argument can be accessed by *varargin{i}*.

# Log with Variable Length Argument

## Code

- For example, a new log function can be defined so that it returns a vector if more than one input is provided.

```
1 function z = log(x, varargin)
2   if nargin == 1
3     z = log(x);
4   else
5     z = [log(x) zeros(1, nargin - 1)];
6     for t = 2:nargin
7       z(t) = log(varargin{t - 1});
8     end; end; end
```

# Output Arguments

## Code

- *varargout* represents an arbitrary number of output variables.
- *nargout* represents the number of output variables assigned when the function is called.
- For example,  $x = \text{size}([1 \ 2; 3 \ 4])$  assigns  $x$  the value **2 2** and  $[x, y] = \text{size}([1 \ 2; 3 \ 4])$  assigns  $x$  the value **2**.

# Recursion

## Math

- A function that uses itself in the body is called a recursive function.

- 1 *function*  $z = f(x)$
- 2 *if*  $x \dots$  % base case
- 3  $z = \dots$
- 4 *else* % recursion
- 5  $z = \dots f(x') \dots$
- 6 *end*

# Recursion Example, Factorial

## Code

- To compute the factorial of  $n \geq 0$ :

```
① function z = f(x)
②   if ~x
③     z = 1;
④   else
⑤     z = x * f(x - 1);
⑥   end
⑦ end
```

# Recursion Example, Vector Sum

## Code

- To compute the sum of the values in a vector  $v$ :

```
1 function z = f(x, t)
2   if nargin == 1
3     z = f(x, 0);
4   elseif t > length(x)
5     z = 0;
6   else
7     z = x(t) + f(x, t + 1);
8   end
9 end
```

# Recursion Quiz Questions

## Quiz

# Blank Slide